Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5540--02-8600

A Methodology, a Language, and a Tool to Provide Information Security Assurance Arguments

JOON PARK
ANDREW MOORE
BRUCE MONTROSE
BETH STROHMAYER
JUDITH FROSCHER

Center for High Assurance Systems
Information Technology Division

February 15, 2002

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE February 15, 2002	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE A Methodology, a Language, and a Tool to Provide Information Security Assurance Arguments	5. FUNDING NUMBERS 55-7287-A-2 PE - RDTE WU - 55-7287
6. AUTHOR(S) Joon Park, Andrew Moore, Bruce Montrose, Beth Strohmayer, and Judith Froscher	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375-5320	8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/5540--02-8600
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Security Agency Ft. George G. Meade, MD 20755	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

As information systems become more complex and industry and military rely more on their correct operation, the need for survivable, secure systems becomes more pressing. System designers and assessors need to clearly understand the causality, relationships, vulnerabilities, threats, system-level view points, and objectives of an entire enterprise. To design a system that can be trusted or assess security properties in a system, the related assurance arguments need to be developed and described effectively in a well-organized format by means of a sound language. To satisfy this requirement, we introduce a methodology, ECM (Enterprise Certification Methodology), to derive and organize the related assurance arguments effectively. We have developed a visual language, CAML (Composite Assurance Mapping Language), to build the map of the assurance argument using ECM. This map depicts the claim trees for the assurance arguments related to the enterprise security objective. We have also developed a tool, VRNM (Visual Network Rating Methodology), to help users develop a map to assurance arguments in CAML based on ECM and document it with related descriptions in a common environment.

14. SUBJECT TERMS Information Security Assurance	15. NUMBER OF PAGES 17
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

CONTENTS

A Methodology, a Language, and a Tool to Provide Information Security Assurance Arguments

Joon S. Park, Andrew Moore, Bruce Montrose, Beth Strohmayer, and Judith N. Froscher
{jpark, moore, montrose, strohmayer, froscher}@itd.nrl.navy.mil

Center for High Assurance Computer Systems
Naval Research Laboratory

Abstract

As information systems become more complex and industry and military rely more on their correct operation, the need for survivable, secure systems becomes more pressing. System designers and assessors need to clearly understand the causality, relationships, vulnerabilities, threats, system-level viewpoints, and objectives of an entire enterprise. To design a system that can be trusted or assess security properties in a system, the related assurance arguments[1] *need to be developed and described effectively in a well-organized format by means of a sound language. To satisfy this requirement, we introduce a methodology, ECM (Enterprise Certification Methodology), to derive and organize the related assurance arguments effectively. We have developed a visual language, CAML (Composite Assurance Mapping Language), to build the map of the assurance argument using ECM. This map depicts the claim trees for the assurance arguments related to the enterprise security objective. We have also developed a tool, VNRM (Visual Network Rating Methodology), to help users develop a map to assurance arguments in CAML based on ECM and document it with related descriptions in a common environment.*

1. Introduction

Today, information technology plays a principal role in industry, government, military, and even citizenry. The emergence of new information technology has changed life styles and the world economy. Information Assurance (IA) problems have become more important as the world uses more information systems, since the security and survivability of information systems directly or indirectly affect organizations, which rely on information technology. Particularly, when we design or assess a large enterprise where many components are integrated with and dependent on each other, it is difficult to extract the necessary assurance arguments for components and describe their relationships and dependencies. This requires that designers (developers) and assessors (evaluators) need to clearly understand the problems at hand and describe them.

[1] In this paper, we use the terms *information security assurance arguments*, *assurance arguments*, and *security arguments* interchangeably.

Manuscript approved December 27, 2001.

Hence, it is obvious that we need effective and efficient methods for system design and assessment, providing causality, relationships, vulnerabilities[2], threats[3], system-level viewpoints, and objectives of an entire enterprise. To design a trusted system or assess security properties in a system, the related assurance arguments need to be developed and described efficiently in a well-organized format by means of an easy-to-use language based on a sound methodology so that decision makers can make informed decisions. This can reduce lifecycle cost through early assurance discussions between designers and assessors, and reduce the need for expensive backtracking.

As information systems become more complex and industry and military rely more on these systems, the need for good IA approaches increase. A useful approach to enterprise assurance would provide the following capabilities:

- A rigorous technique for specifying the security posture for the entire enterprise
- Use of the same representation scheme for reasoning about the enterprise security objective as about a security critical component
- A way for illustrating and organizing the contributions of various types of assurance evidence (e.g., formal specification, security features testing, pseudo-code, penetration testing, etc.)
- A framework for making trade-off decisions, including trade-offs among different disciplines
- A means for understanding the consequences of a change deep in an enterprise assurance argument

In this paper, we introduce three contributions to satisfy the above requirements. Firstly, we have developed a methodology, ECM (Enterprise Certification Methodology), to derive and organize the related assurance arguments effectively. Secondly, we have developed a visual language, CAML (Composite Assurance Mapping Language), to build a map of assurance arguments. This map depicts the claim trees for the assurance arguments related to the enterprise security objective, providing causality, relationships, vulnerabilities, threats, and other system and environment related issues. Finally, we developed a tool, VNRM (Visual Network Rating Methodology), to help users develop a map to assurance arguments in CAML and document it with related descriptions in a common environment. The VNRM tool is not a risk[4] management tool, but rather it provides an environment for applying the ECM and supplies inputs to decision makers by means of assurance argument maps for secure system design and evaluation. Our primary goal is to capture the semantics of a target system's assurance arguments.

This paper is organized as follows. In Section 2, we briefly describe the existing technologies related to our work. In Section 3, we introduce the methodology (ECM) that we use to derive and organize the related assurance arguments for a given enterprise security objective effectively in an assurance map. Section 4 introduces the language (CAML) that we developed to express assurance arguments visually and effectively. Section 5 describes the tool (VNRM) that we developed to help users map assurance arguments in CAML based on ECM. Section 6 introduces

[2] A vulnerability refers to a weakness of a system that can be exploited to undermine the availability, integrity, confidentiality, and/or authenticity of the information resources and/or the system itself.
[3] A threat refers to a circumstance, thing, person, or event - which may occur independent of the system - with the potential to violate the system security or assets.
[4] A risk is a cost estimate based on the probability of a successful attack and the value of the vulnerable asset. For instance, when the probability of successful attack is high and the value of the vulnerable asset is high, the risk is high, and vice versa.

examples of assurance argument maps developed in CAML based on ECM using VNRM. Finally, Section 7 concludes this paper giving a summary and future work.

2. Background

The National Security Agency (NSA) developed a matrix based methodology, Network Rating Methodology (NRM, [BGB97]), an extension of their previous work, Network Rating Model [LBB96], for assessing and evaluating network security, either in operation or in development, based upon a defined set of characteristics. NRM notation is based on a matrix representation. Each row represents an area of security concerns, such as confidentiality, integrity, availability, and authenticity. Each column represents a security service, such as personnel, operational procedure, technology, and physical environment. Each cell is filled with corresponding claims or evidence for the related area of security concern and service. The matrix is multi-dimensional depending on the granularity required by the system owner. In other words, lists of evidence in the next level of the matrix are given to support the first-level claims.

Researchers at the Naval Research Laboratory introduced the assurance strategy [PFL93, FNH94] in 1993. The assurance strategy concerns not only the components (hardware and software) of a system but also the environment in which the components operate. They identify assumptions and assertions that reflect information security requirements for systems and use those concepts to document the trade-off decision. According to their identification, assertions are predicates that are enforced by the system, while assumptions are predicates that are enforced in the system's environment. The assurance strategy initially documents the set of assumptions and assertions derived from the requirements. It is elaborated and refined throughout the development, yielding the assurance argument, delivered with the system, which provides the primary technical basis for the certification decision. If assumptions for any discipline are identified that do not correspond to assertions for some other entity, then these assumptions represent vulnerabilities in using the system.

Systems engineers use fault trees [RVH81] to reason about the safety of composite safety critical systems. Fault trees are a graphic representation of a combination of events that can cause the undesired event to occur. They are developed for analyzing system dependability. In this approach, all the components and their failure probabilities are represented, which consequently cause the failure of the whole system. An event at level l_i is reduced to the combination of lower levels, through some logic ports. Researchers at the University of York and Defense Research Agency (DRA) applied the fault tree approach to the Goal Structuring Notation (GSN, [WKC97]), providing the breakdown of safety requirements to arguments based upon available evidence, to describe the safety arguments and safety assessments. They developed a PC based tool, Safety Argument Manager (SAM, [WMKF96]), which supports GSN.

Researchers at the University of Virginia (UVA) developed the Methodically Organized Argument Tree (MOAT, [KW96]). MOAT provides hierarchically organized arguments in a tree for each security property that is to be analyzed. Each MOAT encapsulates an argument that the system exhibits some desired security property. The MOAT representation looks similar to fault trees. However, the MOAT methodology permits the integration of existing security assurance techniques, such as formal methods, into a risk-driven process model. It orders the construction of assurance arguments using a high-risk-first heuristic, and explores higher-risk areas more fully and precisely than lower-risk areas.

Although the original purposes of GNS, MOAT, and CAML (described in Section 4) are different, we have combined and extended the GNS and MOAT specifications to develop CAML notations, which support our own methodology, ECM (described in Section 3).

3. Enterprise Certification Methodology (ECM)

This section describes the Enterprise Certification Methodology (ECM) for deriving related assurance arguments for a given security objective and organizing them effectively. There is always more than one way to provide security services for an enterprise, and there may be many different methodologies and styles for describing information security assurance arguments for the same information system. This is analogous to a different person describing the same subject with different format and style in English. Even the same person may describe it in a different way from his own prior description. Each description may be acceptable as long as it uses correct English vocabulary, syntax, and grammar and makes a convincing description about the subject. Usually, however, there are some exemplary patterns of description for different purposes. For instance, we have writing formats for poems, articles, reports, novels, etc. for better expression and structure in literature.

Similarly, to provide a good methodology for users to produce an efficient and comprehensive description of information security assurance arguments, we organize the arguments in four different disciplines in a map based on the NRM approach (described in Section 2): physical, personnel, technical, and operational. This map shows the claim trees for the assurance arguments related to the objective, providing causality, relationships, vulnerability, threats, and other system and environment related issues. We do not propose to "hard code" the precise partitioning of the four disciplines in the argument maps for every application. Different systems or applications may have a different partitioning of their arguments for improved understandability or a different level of assurance. However, we believe the four different disciplines provide comprehensive categories for most applications or systems.

Physical Security involves the strength of physical mechanisms and structures used to protect and house the technology, such as strength of locks or safes. Operational Security involves the effectiveness of manual procedures, policies, and guidelines for handling and protecting information. The more conventional view of assurance comes from Technological Security, which involves security about combinations of hardware, software, and communications. Finally, Personnel Security involves assurance about people, their trustworthiness and capabilities through some processes, such as personnel background investigation, training, and evaluation.

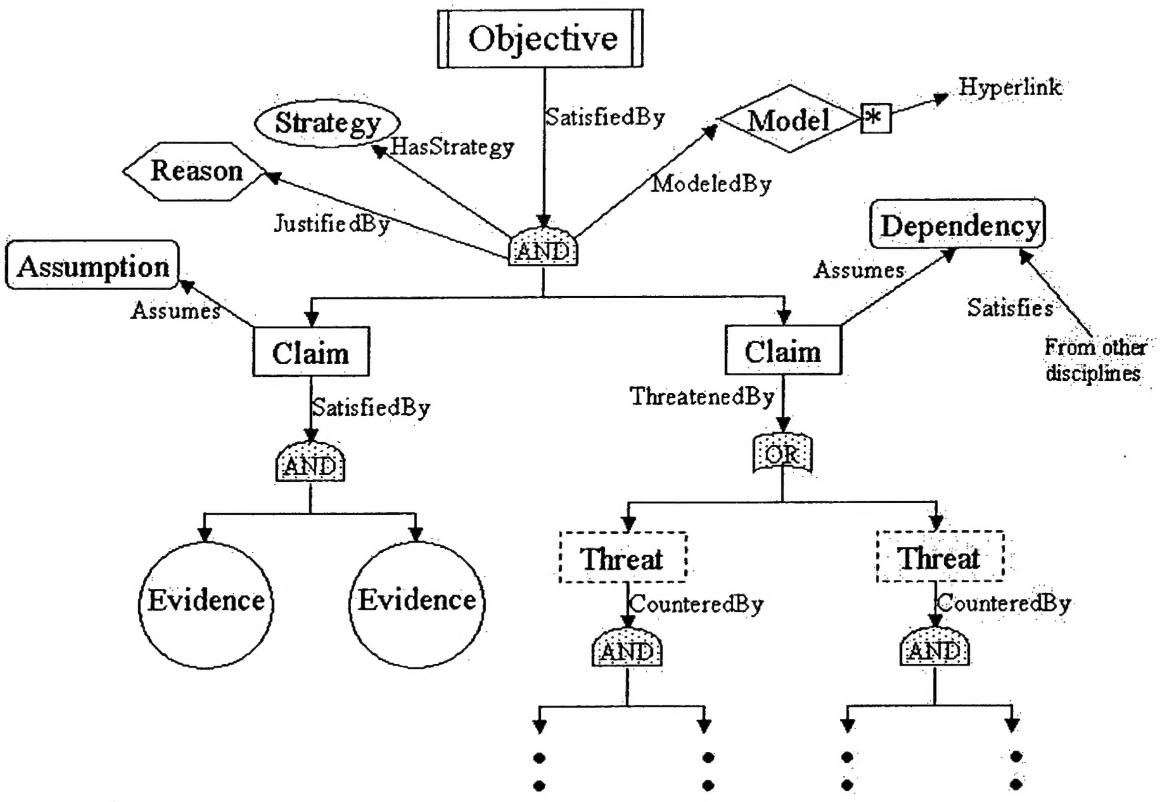
Partitioning security requirements into the four security disciplines helps users derive the related security arguments comprehensively in each discipline and integrate them together, providing their relationships and causality. This approach also helps determine how to protect information in the most cost-effective manner. Since we cannot practically avoid risk altogether, the task becomes one of developing a partitioning that provides a good balance between affordability and risk reduction. One could, for instance, introduce guard inspections into a system to reduce the risk of physical threats. This would, of course, increase the personnel and operational costs associated with operating the system. The program manager would need to decide whether the reduction in physical risk is worth the corresponding increase in personnel and operational costs.

Moreover, we extend and apply our assurance strategy (described in Section 2) to the four security disciplines in a CAML map (described in Section 4) to elaborate the high level enterprise security objectives. The CAML map can be used to develop a rigorous specification of the security posture of the entire enterprise. As a result, a CAML map permits tracking security vulnerabilities by identifying assumptions made in one discipline of the map that are not matched by validating claims made in another discipline of the map. In other words, a statement can be an assumption in one discipline while it can be a claim in another discipline. For instance, the statement "Users know authentication procedures" is an assumption under the technical discipline, while it is a claim under the operational discipline. Such assumptions become *dependencies* of the argument. Assumptions that are not so linked become vulnerabilities that need to be considered when assessing residual risk. These vulnerabilities must be assessed when deciding whether the residual risk is tolerable in the operational environment. The evaluator also needs to ensure the integrity of the assumption validation mapping itself. If a portion of the map for one application is reused for another application, the reader may have noticed the lack of any claims to ensure consistent application of the assumptions. Conscientious application of the above approach helps to uncover such gaps, identifying security vulnerabilities that were not previously considered.

4. The Composite Assurance Mapping Language (CAML)

We have developed CAML by merging and extending several existing technologies (described in Section 2) to describe information security assurance arguments effectively in a well-organized format. In this section, we briefly describe the rules and components of CAML. The detailed description and usage of CAML is available in [MMS00]. Figure 1 shows an example of a CAML structure with its primitives and definitions.

Distinct graphical primitives in different shapes represent key components of the argument map. A textual summary of each component is shown inside each shape. The spine of an argument map hierarchically refines security *claims* about the system into sub-claims that, eventually, are linked with the *evidence* that a claim is satisfied. The flesh of an argument map describes supporting information about the refinement such as the general *strategy*, *assumptions* and *dependencies* made, justifying *reasons*, and contextual *models*. Spine refinement may proceed using either *AND*-decomposition or *OR*-decomposition. By the AND-decomposition, all sub-claims or evidence must hold for the decomposed claim to hold. By the OR-decomposition, one of the sub-claims or evidences must hold for the decomposed claim to hold.



Definitions

- Objective: A statement expressing a security requirement of the countermeasure, system, network or enterprise that is the reference of an argument.
- Claim: Statements that associate subjects with their attributes or properties.
- Assumption: A claim that is accepted without justification.
- Dependency: An assumption in one part of an argument that is validated by a claim in another part of the argument.
- Evidence: Data on which a judgment or conclusion about an assurance claim may be based.
- Hyperlink: A link from one component of an argument map to other components of the map or external components to provide for the detail or clarification to the argument.
- Strategy: The approach taken for refining a claim into sub-claims or into evidences supporting the claim.
- Reason: A set of statements that ties together a set of sub-claims or evidences to establish a claim.
- Model: The architectural context on which a claim decomposition is based.
- Threat: A statement that expresses any circumstance or event with the potential to cause harm to an asset.

[Figure 1] An example of a CAML structure with its primitives and definitions

Not all CAML components are needed for every assurance map. Map developers use their discretion for choosing the necessary components to convey their argument satisfactorily. When a flesh component is connected to an AND/OR connector, it means this flesh component applies to all the arguments below the AND/OR connector. When a flesh component is connected to a spine (claim or evidence) directly, it means the flesh component applies to the particular spine. To provide more detailed descriptions of the map, the shapes can be hyperlinked. For instance, architectural diagrams can be hyperlinked to model shapes, and analytic proofs and tests can be hyperlinked to evidence shapes.

The claim trees created and denoted by CAML provide two characteristics that form a component's security interface with other components and other IA disciplines. The first characteristic is the set of *claims* made about the component's security functionality. In CAML, there can be many claims that portray refinements of more abstract claims for the purpose of providing an assurance map for a specific component. From a system composition point of view, we are interested in the claims about the component's security functions that may be referred for other components. The second characteristic is a set of *assumptions* about other components or other IA disciplines, that must be true in order for the component claims to be true. The assumptions that form part of the component's security interface are those that are made about components and IA disciplines external to the component over which it has no control. To compose a secure system, all of the external assumptions upon which the claims of one component depend must be supported by a claim that can be made about another component or IA discipline that have strong evidence of their truth. Failure to satisfy this composition rule, identifies vulnerabilities - a residual risk or a hole in the protection layer. In order for this process to work, all the components in the system would need a security interface described in terms of claims and assumptions. Claims must be supported by assurance evidence and /or assumptions. Components that are not trusted would not require assurance arguments and one may only make assumptions about their security related functionality. Assumptions about components and IA disciplines over which the component has control, usually, remain vulnerabilities. Detailed methodologies to build a CAML map are described in Section 3.

Claim trees describing individual components will vary in their level of refinement and strength of evidence. Some products may have very weak claims that are only supported by internal assumptions (assumptions about their internal parts). Others may have layers of refinement with hyperlinks to substantial evidence that the lower layers are true. Documentation for these components may be many pages in length. At the system level, only the external claims and assumptions (the component's security interface) need to be shown. Lower levels of refinement should be merely referenced.

CAML provides a medium through which to develop a concise roadmap of the evidence that an enterprise conforms to required security properties. Resolving the competing factors involved with secure enterprise design typically requires depending on diverse assurance that need to be interrelated in complex ways. Users coherently and consistently map out the assurance evidence and its associated reasoning using CAML maps. The language helps to tame the complex inter-relationships of the diverse sources from which that evidence originates, and to illuminate the process of balancing mission, security, and cost. Such a common language is critical as a communication medium for risk analysts, developers and evaluators. It will allow them to come to a consistent understanding and agreement on assurance requirements and how an enterprise satisfies those requirements. Well-constructed CAML maps contain rationale that promotes confidence in their own consistency and completeness. Additionally, getting early evaluator feedback in the assurance mapping process helps to ensure that the final enterprise design will be approved for operation with minimal changes.

5. Visual Network Rating Methodology (VNRM)

We have developed a toolset, VNRM, to help users draw a graphical assurance argument map in CAML (described in Section 4) base on ECM (described in Section 3) that evaluates if the target system adequately supports security services. The VNRM User's manual [MS00] is helpful in getting one started on using the tool. Additional information, including a packaged demonstration, is available at the VNRM website [Vis00].

The VNRM toolset was built to be easy to use, maintain, and extend. Its design supports ease-of-use by using graphical user interfaces that are familiar to a large portion of the potential user community. The design uses standard, widely accepted software components that support both the need for familiar interfaces and the need to maintain compatibility of VNRM with cutting edge technology. As well-supported software components evolve, VNRM can evolve in like manner with a minimal amount of effort. Finally, extensibility is important to the evolving VNRM design so that when users identify additional, value-added functions, the toolset can be extended easily with minimal changes to the existing implementation. The design supports extensibility primarily through the use of client-server and modular design techniques.

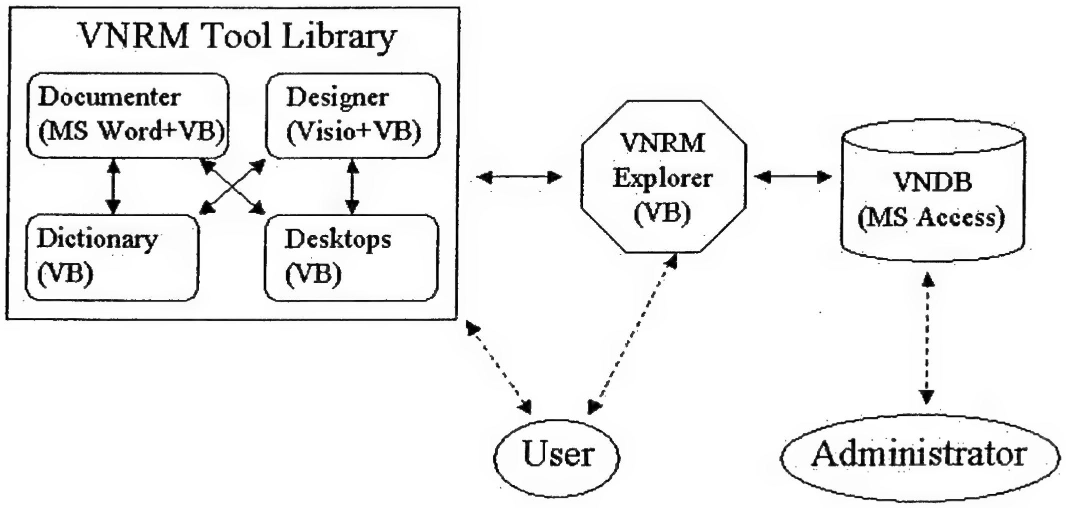


Figure 2. VNRM tool architecture

Figure 2 depicts the VNRM tool architecture. Solid lines represent invocation and message passing between VNRM components and dotted lines represent interactions between human beings and VNRM components. The VNRM tool consists of three major components: VNRM Explorer, VNRM Tool Library, and VNRM Database (VNDB). The VNRM Explorer, an interface that has the familiar look and feel of the Microsoft Windows Explorer, provides a user-friendly front-end to the VNRM Database (VNDB) for tools in the VNRM Tool Library. It provides a utility to help manage, build, and review CAML assurance argument maps. The VNDB, which is implemented in Microsoft Access, stores the artifacts of an assurance argument, including the CAML map and its documentation, on a project-by-project basis. Users manage and access VNRM projects through the VNRM Explorer. Tools in the library can update the VNDB only through the VNRM Explorer so as to preserve the consistency of the VNDB data and the tools' views of those data. The VNRM Explorer notifies any tools that have a need to know when data is updated.

Four tools that support creating and documenting CAML argument maps currently reside in the VNRM Tool Library. VNRM Designer uses the Visio extensible drawing package to create, analyze, and hyperlink CAML maps. These maps can be integrated (as OLE links) into textual documents using the VNRM Documenter, which is implemented using Microsoft Word. Both the Visio and Word environments were extended using Visual Basic (VB) to support VNRM-specific function. The VNRM Dictionary permits defining a standard terminology for consistent application across or within VNRM projects. Terms so defined are highlighted in the textual parts of the map and its documentation. Finally, VNRM Desktops provide a virtual desktop - which provide collections of pages that are developed, analyzed, and/or presented as a group - function to associate different segments of an assurance argument map for simultaneous elaboration or examination.

VNRM helps manage the complexity of composition of diverse enterprise assurance arguments by mapping out the assurance evidence, tracing meaningful threads of reasoning, and highlighting significant results. Taking an enterprise view of assurance gives us many places from which to derive confidence that information security is not compromised. It clearly shows how each piece of evidence in the map contributes to the overall argument. Furthermore, it supports traversing the hyperlinked argument and analyzing its security weakness. Documenting this rationale is important for understanding why key decisions were made and the impact modifying the enterprise design has on information assurance.

Another purpose of this tool is to help users identify residual vulnerabilities and experiment with moving them around not only the system components, but around the whole enterprise. This helps improve security, by keeping the vulnerability away from the threat agent[5], and permits developing affordable solutions to information security. By making tradeoffs between responsibilities of different security disciplines, we can exchange costs and risks as appropriate for the system under construction.

VNRM can be used for a risk analysis process, an engineering process, and a security evaluation process to develop a secure enterprise. Early risk analysis involves identifying both natural and man-made threats to enterprise assets, and the corresponding consequence to the enterprise if those assets are compromised. This analysis results in a set of protection needs that provide a starting point for assurance mapping analysis. We translate these protection needs into a set of high-level claims to which the enterprise must conform by considering the role that the enterprise plays in meeting the protection needs. A primary goal of an assurance map is to show how and

[5] The person, organization, or circumstance that implements a threat.

why these claims are satisfied. Assumptions on which this argument is based indicate potential enterprise vulnerabilities, to the extent that if not properly validated they indicate a hole in the enterprise assurance argument. The strength of the reasoning and evidence on which the argument is based indicates the confidence that we have identified existing vulnerabilities in the map. Such vulnerability analysis is a critical component of the risk analysis process. Of course, the engineering process must consider other factors in addition to information security during enterprise design. Security protection must be balanced within the overall mission of the enterprise, such as with the costs associated with development and operation. The way that the user allocates security responsibilities to different disciplines emphasizes certain factors over others.

In summary, the VNRM tool has the following functions.

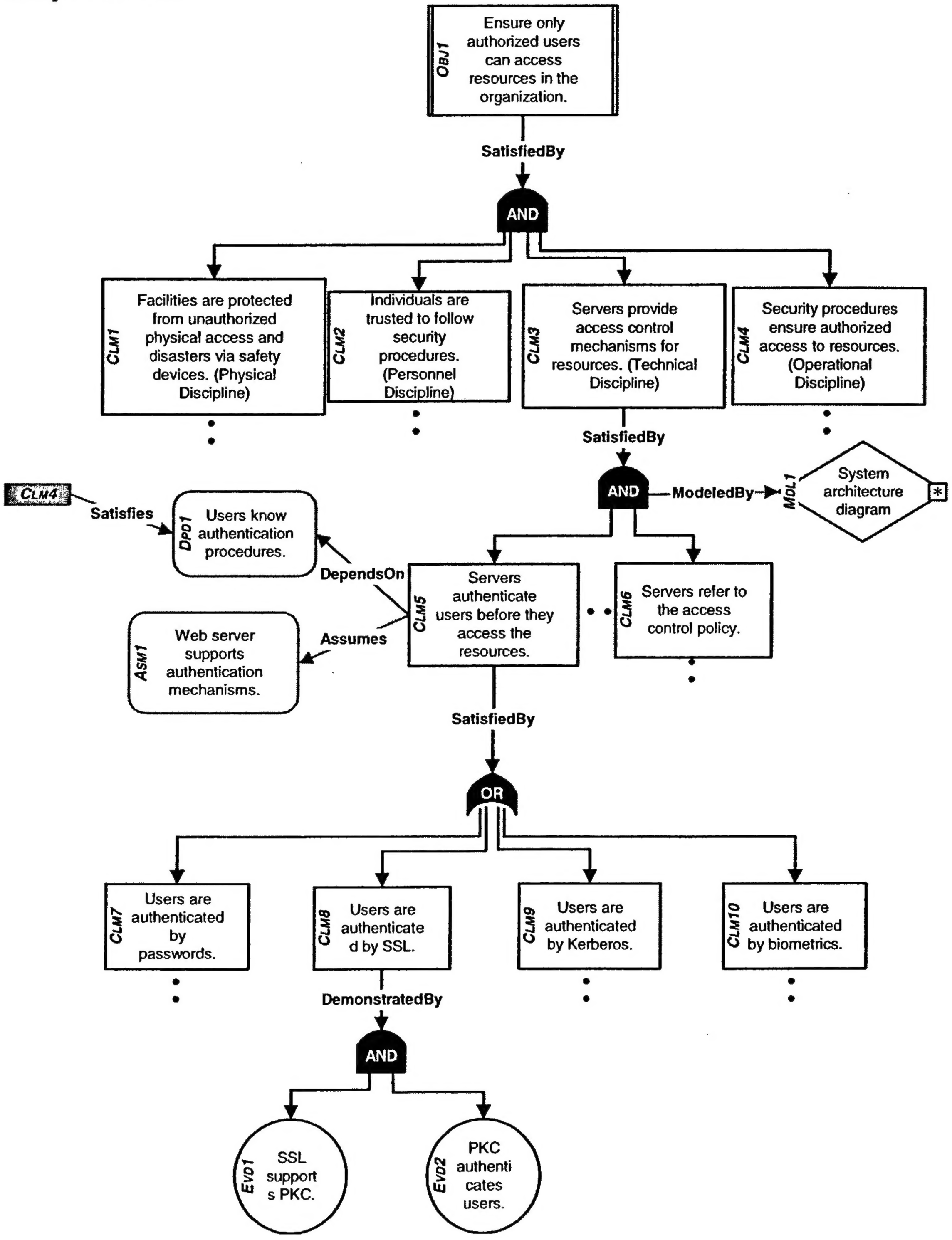
- Construct a comprehensive roadmap of evidences that the enterprise, system, or component satisfies required security properties
- Integrate and build assurance maps from well-defined composable building blocks
- Provide hyperlinks to the assurance arguments for detailed description
- Discover and enumerate vulnerabilities
- Document security design rationale and tradeoffs

The above functions of VNRM provide the following benefits.

- Better understanding of when, how, and why security components and controls can be composed and the aggregate assurance that results
- Improved information security design by having methods available to help determine the trade-off between alternative designs with respect to mitigating security vulnerabilities
- A common language and environment for enterprise stakeholders to communicate
- Graphical presentation promotes understanding and analysis of system
- Expanded capability to reason about the strength and effectiveness of different combinations of security services and policies
- Reduced lifecycle cost by enabling reuse of independently developed components and their assurance arguments in composite systems
- Freedom to use a degree of rigor chosen by the developer as appropriate for the application of interest
- Enhanced maintainability by having composite argument well-documented and tools available to help predict the impact of modifications to the argument and suggest ways to compensate
- Increased objectivity of system evaluation and accreditation decision

6. Examples of CAML Maps Developed Using VNRM

To prove the feasibility of our work, we released VNRM to evaluate real applications. The DARPA IA Program's Assurance Working Group (AWG) has used VNRM to characterize the strength and effectiveness of two different security technologies: the Object-Oriented Domain-Type Enforcement (OO-DTE, [STM99]) and the Advanced Research Guard for Experimentation (ARGuE, [Eps99]). These efforts have largely been successful, as seen by the growing appreciation of the techniques on which VNRM is based. VNRM is currently being applied to the integration of DARPA IA technologies to implement a Virtual Private Network (VPN) in an enterprise context.



OBJ1
Ensure only authorized users can access resources in the organization.
SatisfiedBy
AND
CLM1
Facilities are protected from unauthorized physical access and disasters via safety devices. (Physical Discipline)
CLM2
Individuals are trusted to follow security procedures. (Personnel Discipline)
CLM3
Servers provide access control mechanisms for resources. (Technical Discipline)
CLM4
Security procedures ensure authorized access to resources. (Operational Discipline)
SatisfiedBy
AND
ModeledBy
MDL1
System architecture diagram
CLM4
Satisfies
DPD1
Users know authentication procedures.
DependsOn
ASM1
Web server supports authentication mechanisms.
Assumes
CLM5
Servers authenticate users before they access the resources.
CLM6
Servers refer to the access control policy.
SatisfiedBy
OR
CLM7
Users are authenticated by passwords.
CLM8
Users are authenticated by SSL.
CLM9
Users are authenticated by Kerberos.
CLM10
Users are authenticated by biometrics.
DemonstratedBy
AND
EVD1
SSL supports PKC.
EVD2
PKC authenticates users.

Figure 3. An example of CAML map developed using VNRM

In this section, we show an example of a CAML map (denoted in Figure 3) that we developed using VNRM based on ECM to illustrate the approach. Because of space constraint, we captured only part of the actual work in this paper. The black dots around the claim boxes represent the omitted parts in the figure. In the original VNRM environment, distinct graphical primitives are denoted in different colors as well as in different shapes. Note that a different person may develop a different assurance map from others for the same system. Although VNRM has other functions, such as documenter, dictionary, and desktops, we focus only on the example CAML map in this paper.

The purpose of this map is to show the arguments related to the objective, "Ensure only authorized users can access resources in the organization." As we introduced in Section 3, we partition the related security arguments into the four disciplines: physical, personnel, technical, and operational security disciplines. Each discipline has a high level claim (CLM1 through CLM4 in the map), which is satisfied by sub-claims and related arguments. Those claims are connected by an AND-connector because all those claims must hold for the decomposed objective to be satisfied. The decompositions for physical, personnel, and operational disciplines are omitted in this particular example. The claim for physical security discipline (CLM1) would be decomposed with arguments related to physical entry to the facility, storage cabinets, safety devices, etc. The claim for personnel security discipline (CLM2) would be decomposed with arguments related to users, administrators, guards, etc. The claim for operational security discipline (CLM4) would be decomposed with arguments related to organizational policies, guidelines, etc.

In this example, we decompose the claim for the technical security discipline (CLM3). A system architecture diagram is provided by a Model shape (MDL1) that is hyperlinked (denoted by * symbol in MDL1) to external information (not shown in this paper). Technically, the servers authenticate users before they access the resources (CLM5), where we assume that users know the authentication procedures and Web server supports authentication mechanisms. The former assumption is denoted in a Dependency shape (DSD1) because it has a verifying link[6] (CLM4) to a different partition (operational discipline), while the latter assumption is denoted in an Assumption shape (ASM1), which still shows vulnerabilities, because it does not have verifying links (described in Section 3). Additionally, the servers refer to the access control policy of the organization to make an access control decision (CLM6). Since CLM5 through CLM6 are required to hold for CLM3 to be true, they are connected by an AND-connector. Continuing the decomposition, CLM5 is decomposed into four possible authentication techniques (CLM7 through CLM10) by means of an OR-connector. This means that one of those authentication techniques must be provided. Based on a given situation, we may add more authentication techniques under the OR-connector or remove some of them. Finally, the claim CLM8 (Users are authenticated by SSL) is decomposed in two evidences via an AND-connector. Those evidences demonstrate why we can trust that SSL can authenticate users. Similarly, the other claims can be decomposed into evidence.

[6] If the claim for the operational security discipline would be decomposed more in this example, this assumption would be linked to more refined claims.

7. Conclusions and Future Work

We have developed a methodology (ECM), a language (CAML), and a tool set (VNRM) to provide a capability for rigorously specifying the security posture and assurance map of an enterprise by merging and extending several existing technologies to describe information security assurance arguments effectively in well-organized formats. VNRM helps users develop a graphical argument map in CAML based on ECM and document it with related descriptions. The VNRM toolset was built to be easy to use, maintain, and extend.

Usually, large-scale application of assurance approaches to non-trivial enterprise integration cannot proceed solely from high-level mission and security objectives, but instead follow from the design of enterprise computing resources. As with any system development process, an engineer must be able to identify the reusable elements and compose these elements into useful portions of the enterprise. A popular approach to specifying reusable elements of system design is by specifying design patterns; abstractions from a concrete recurring solution that solves a problem in a certain context. Future work will investigate ways to define a notation and theory for specifying and composing argument patterns and will create a repository of previously developed arguments that can be shared and reused. We will catalog these patterns and populate a repository for assembling different combinations of mechanisms and their composite assurance arguments. Users will be able to construct and share argument patterns and instantiate and compose them into full-fledged assurance arguments for composite systems.

References

[BGB97] R. Bailey, B. George. C. Bowers, et al. *The Network Rating Methodology: a Framework for Assessing Network Security*, http://chacs.nrl.navy.mil/projects/VisualNRM /nrm_3rd_draft.html, National Security Agency, 1997.

[Eps99] J. Epstein. *Architecture and Concepts of the ARGuE Guard*. Proceedings of 15th Annual Computer Security Applications Conference (ACSAC), Phoenix, Arizona, December 1999.

[FNH94] J. Freeman, R. Neely, and M. Heckard. *A Valid Security Policy Modeling Approach*. Proceedings of 10th Annual Computer Security Applications Conference (ACSAC), Orlando, Florida, December 1994.

[KW96] W. Wulf, C. Wang, and D. Kienzle. *A New Model of Security for Distributed Systems*. Proceedings of the New Paradigms in Security Workshop, Lake Arrowhead, California, 1996.

[LBB96] O. Lambros, R. Bailey, C. Bowers, et al. *Network Rating Model: An Approach for Assessing Network Security*, http://www.radium.ncsc.mil/nrm/rev961031.html, National Security Agency, 1996.

[MMS00] A. Moore, Bruce Montrose, and Beth Strohmayer. *A Tool for Mapping Enterprise Security Assurance*. Technical Report 5540-051a:apm, Naval Research Laboratory, September 2000.

[MS00] A. Moore and B. Strohmayer. *Visual NRM User's Mannual*. Technical Report NRL/FR/5540--00-9950, Naval Research Laboratory, May 2000.

[PFL93] C.N. Payne, J.N Froscher, and C.E. Landwehr. *Toward a Comprehensive INFOSEC Certification Methodology*. Proceedings of 16th National Computer Security Conference (NCSC), pages 165-172, Baltimore, MD, September 1993.

[RVH81] N. Roberts, W. Vesely, and D. Haasl. *Fault Tree Handbook*. NUREG-0492, Office of Nuclear Regulatory Research, 1981.

[STM99] D. Sterne, G. Tally, C. McDonell, et al. *Scalable Access Control for Distributed Object Systems*. Proceedings of 8th USENIX Security Symposium, Washington, D.C., August 1999.

[Vis00] *VNRM (Visual Network Rating Methodology): Tools for Mapping Assurance Arguments*. http://chacs.nrl.navy.mil/projects/VisualNRM/, Naval Research Laboratory, 2000.

[WKC97] S. Wilson, P. Kirkham, and M. Cassano. *SAM 4 User Manual*. University of York, 1997.

[WMKF96] S. Wilson, J. McDermid, P. Kirkham, and P. Fenelon. *The Safety Argument Manager: An Integrated Approach to the Engineering and Safety Assessment of Computer Based Systems*. Proceedings of IEEE Symposium and Workshop on Engineering of Computer-Based Systems, page(s): 198 –205, 1996.